DRAFT Specification

IDA System Interface (ISI)

David Chavez

Introduction

The IDA System Interface is being developed as the standard for communication between IDA seismic stations, IDA data collection hubs, and IDA TCP/IP enabled digitizers. It uses the IDA Authenticated Communication Protocol (IACP) as the underlying transport. This document is limited to a description of the IACP frames that have been assigned to the ISI.

The ISI implementation presently supports state of health, configuration, and waveform transfer between IDA/NRTS computers. Client side source code is available for pickup at

ftp://idahub.ucsd.edu/pub/pickup/ISI_Toolkit.tgz

This software includes library routines for conducting dialogs with an ISI server and sample programs that demonstrate the use of those routines.

ISI Session Overview

An ISI session consists of establishing an IACP connection to the ISI server and then sending ISI frames and reading server responses as ISI frames. The simplest session would be one where the client connects to the ISI server, issues a state of health, configuration, or disk loop summary report, receives the reply, processes it (e.g., displays it or writes it to disk) and then closes the connection. Waveform sessions are more involved. They require that the client send a series of frames that constitutes the waveform request, then read a series of waveform frames in reply. This document discusses the IACP payload format that are used to conduct all such sessions.

ISI Payload Types

IACP payload identifiers 1000 through 1999 have been assigned for use by ISI. Presently, the following 12 types have been defined:

```
#define ISI_IACP_REQ_SOH      1001 /* state of health request */
#define ISI_IACP_REQ_CNF      1002 /* configuration request */
#define ISI_IACP_REQ_WFDISC   1003 /* disk loop summary request */
#define ISI_IACP_REQ_FORMAT   1004 /* format part of data request */
#define ISI_IACP_REQ_COMPRESS 1005 /* compress part of data request */
#define ISI_IACP_REQ_POLICY   1006 /* policy part of data request */
#define ISI_IACP_REQ_STREAM   1007 /* stream part of data request */
#define ISI_IACP_SYSTEM_SOH   1008 /* system SOH data (unused) */
#define ISI_IACP_STREAM_SOH   1009 /* stream SOH data */
#define ISI_IACP_WFDISC       1010 /* wfdisc record */
#define ISI_IACP_STREAM_CNF   1011 /* configuration data */
#define ISI_IACP_GENERIC_TS   1012 /* ISI generic time series packet */
```

ISI Data Types

The discussion of the payload formats that follows will refer to the fundamental data types given in the following table.

Name	Length	Description
INT8	1	signed byte
UINT8	1	unsigned byte
INT16	2	signed 16-bit integer, network byte order
UINT16	2	unsigned 16-bit integer, network byte order
INT32	4	signed 32-bit integer, network byte order
UINT32	4	unsigned 32-bit integer, network byte order
INT64	8	signed 64-bit integer, network byte order
UINT64	8	unsigned 32-bit integer, network byte order
REAL32	4	32-bit IEEE floating point, network byte order (cast as UINT32)
REAL64	8	64-bit IEEE floating point, network byte order (cast as UINT64)
CHAR	1	ASCII character

These fundamental data types are used to build more complicated data types.

ISI_STREAM_NAME

An ISI_STREAM_NAME is a 12 byte data structure with the following format:

Offset	Data Type	Length	Description
0	CHAR	7	station name
7	CHAR	3	channel name
10	CHAR	2	location code

ISI_TIME_STAMP

An ISI_TIME_STAMP is used to specify an epoch time. It is a 10 byte data structure with the following format:

Offset	Data Type	Length	Description
0	REAL64	8	Seconds since January 1, 1970
8	UINT16	2	Clock status

ISI_SRATE

An ISI_SRATE describes a sample rate. It is a 4 byte data structure with the following format:

Offset	Data Type	Length	Description
0	INT16	2	Sample rate factor
2	INT16	2	Sample rate modifier

The sample rate factor and modifier fields follow the same convention as used in the SEED specification. If the sample rate factor is positive, it represents samples per second otherwise it represents seconds per sample. In either case, the value is modified by the sample rate modifier field. If the modifier is positive, the factor is multiplied by the modifier otherwise it is divided by the modifier.

ISI_COORDS
An ISI_COORDS is specifies location. It is a 16 byte data structure with the following format:

Offset	Data Type	Length	Description
0	REAL32	4	latitude, decimal degrees
4	REAL32	4	longitude, decimal degrees
8	REAL32	4	surface elevation, meters
12	REAL32	4	depth of burial, meters

ISI_INST
An ISI_INST contains basic information about a particular data stream. It is a 23 byte data structure with the following format:

Offset	Data Type	Length	Description
0	REAL32	4	CSS 3.0 “calib”
4	REAL32	4	CSS 3.0 “calper”
8	REAL32	4	CSS 3.0 “hang”
12	REAL32	4	CSS 3.0 “vang”
16	CHAR	7	CSS 3.0 “instype”

ISI_CD_STATUS
An ISI_CD_STATUS contains CD1.1 channel status (format 1) parameters. It is a 16 byte data structure with the following format:

Offset	Data type	Length	Description
0	UNIT8	1	CD1.1 data status
1	UINT8	1	CD1.1 security
2	UINT8	1	CD1.1 miscellaneous status
3	UINT8	1	CD1.1 voltage status
4	REAL64	8	time of last GPS synchronization
12	UINT32	4	clock differential in microseconds

ISI_DATUM_DESC
An ISI_DATUM_DESC describes the contents of the data portion of a waveform packet. It is a 4 byte data structure with the following format:

Offset	Data type	Length	Description
0	UNIT8	1	Compression 0 = none 1 = IDA first difference 2 = Steim 1 3 = Steim 2
1	UINT8	1	Uncompressed data type 1 = INT8 2 = INT16 3 = INT32 4 = INT64 5 = REAL32 6 = REAL64 11 = native IDA rev 8 raw digitizer packet 12 = native IDA rev 10 raw digitizer packet
2	UINT8	1	Byte Order 0 = little endian byte order 1 = big endian byte order
4	UINT8	1	Uncompressed sample size in bytes

ISI State of Health Request

An ISI state of health report describes the status of the disk loop at the server. This report is obtained by sending a zero length IACP frame with a payload identifier of 1001. The server will reply with a series of frames with IACP payload identifier of 1009. Each type 1009 frame contains state of health information for one data stream. After the final type 1009 frame has been sent, the server will send an IACP NULL frame (payload identifier 0). A type 1009 payload consists of 48 bytes with the following layout:

Offset	Data Type	Length	Description
0	ISI_STREAM_NAME	12	stream name
12	ISI_TIME_STAMP	10	time of oldest datum available on server
22	ISI_TIME_STAMP	10	time of youngest datum available on server
32	REAL64	8	elapsed time since server received data for this stream
40	UINT32	4	number of data segments for this stream
44	UINT32	4	number of data records received by server for this stream

ISI Configuration Request

An ISI configuration report describes the data streams that are available from the server. This report is obtained by sending a zero length IACP frame with payload identifier 1002. The server will reply with a series of frames with IACP payload identifier of 1011. Each type 1011 frame contains configuration information for one data stream. After the final type 1011 frame has been sent, the server will send an IACP NULL frame (payload identifier 0). A type 1011 payload consists of 55 bytes with the following layout:

Offset	Data Type	Length	Description
0	ISI_STREAM_NAME	12	stream name
12	ISI_SRATE	4	sample rate
16	ISI_COORDS	16	coordinates
32	ISI_INST	23	instrument parameters

ISI Disk Loop Summary Request

An ISI disk loop summary is a collection of CSS 3.0 wfdiscs that describes the current state of data in the disk loop. This report is obtained by sending a zero length IACP frame with payload identifier 1003. The server will reply with a series of frames with IACP payload identifier of 1010. Each type 1010 frame contains configuration information for one data stream. After the final type 1010 frame has been sent, the server will send an IACP NULL frame (payload identifier 0). A type 1010 payload consists of 283 bytes with the following layout:

Offset	Data Type	Length	Description
0	CHAR	283	CSS 3.0 wfdisc string

ISI Waveform Request

An ISI waveform request is made by sending a sequence of 3 types of frames that specify the details of the request, terminated by an IACP NULL frame. The 3 frame types are

- 1004 (ISI_IACP_TYPE_REQ_FORMAT) for specifying the delivery format of the waveform packets
- 1005 (ISI_IACP_TYPE_REQ_COMPRESS) for specifying the data compression to apply, if any
- 1007 (ISI_IACP_TYPE_REQ_STREAM) to specify the desired time window for a particular data stream or streams. The details of the payload of each frame follow. *As many instances of the type 1007 frame may be sent as necessary in order to completely specify the list of desired streams*.

ISI_IACP_REQ_FORMAT (type 1004)

This is an IACP frame with payload identifier 1004. The payload is a single UINT32.

Offset	Data Type	Length	Description
0	UINT32	4	0 for generic format 1 for native digitizer packet format

Generic format eliminates any digitizer specific encoding while native format is the raw digitizer packet. In either case, the data are returned in the data field of the generic time series packet (type 1012) described later in this document.

ISI_IACP_REQ_COMPRESS (type 1005)

This is an IACP frame with payload identifier 1005. The payload is a single UINT32.

Offset	Data Type	Length	Description
0	UINT32	4	0 for uncompressed 1 for IDA first difference 2 for Steim 1 3 for Steim 2

This value specifies how the server will compress the waveform data.

ISI_IACP_REQ_STREAM (type 1007)

This is an IACP frame with payload identifier 1007. The payload is a 28 byte structure with the following layout:

Offset	Data Type	Length	Description
0	ISI_STREAM_NAME	12	stream name, possibly with wild cards
12	REAL64	8	time of first datum
20	REAL64	8	time of final datum

The stream name is the (*sta*, *chan*, *loc*) triple used to name the particular data stream. Limited wildcarding is permitted, as follows. If any of the 3 components of the name is given as an asterisk, the server interprets it to mean all values applicable to that server. For example, a triple ("*", "*", "*") would be interpreted to mean all streams available on the server, ("ABC", "*", "*") would mean all streams from station ABC, while ("*", "BHZ", "00") would mean all BHZ00 streams from all stations, etc.

The time of first datum is the epoch time (seconds since January 1, 1970) for the *desired* first sample and the time of final datum is the epoch time of the *desired* final sample. The server will select the waveform packets that contain the desired times, but will not split or otherwise modify the original time span of the packets as they reside on the server. This means that, in practice, the client can expect the first waveform packet to begin slightly before the requested start time and the final packet can extend to slightly beyond the requested time. Note that since epoch time is specified as seconds since January 1, 1970 and it is now past that date, all epoch times will be positive numbers. We take advantage of this fact to define 3 wildcard times that are the following negative values.

Value	Interpretation
-2	oldest packet available
-3	most recent packet available
-4	continuous data feed

The values of -2 and -3 can be used to specify either or both of the time boundaries. The -4 value is only valid for specifying the time of final datum. Any waveform request that does not specify -4 as the time of final datum is a segment request. Such a request spans a specific time interval. Any request which specifies -4 as the time of final datum is a continuous data request. In such a case the server will keep the connection alive indefinitely (with IACP heartbeats if necessary). As new data are acquired the server will forward them to the client.

For example a (begin, end) time pair of (-2, -3) would mean that the client wants to see all the data currently available at the server, while (-2, -4) would mean that the client wants all the data currently available on the server and all additional data the server would obtain in the future. Likewise, (-3, -4) would mean the client has no interest in any old data which are available but wants to get any data acquired since the request was made.

ISI Waveform Protocol

The client initiates a waveform request by sending an IACP NULL terminated series of request frames as described above. The server acknowledges the waveform request by returning the same IACP NULL terminated series of request frames, however it will expand any wild carded ISI_IACP_REQ_STREAM frames into multiple frames with explicit values for the (*sta*, *chan*, *loc*).

After the server has acknowledged the waveform request, it starts sending out generic time series frames (type 1012) that contain the requested data. If the waveform request was a segment request then once the final waveform frame has been sent the server will send an IACP Alert frame (payload identifier 100) with a cause code of 2, indicating that the request has been fully satisfied. Note that this does not mean that all the requested data were actually delivered, it only means that all the data that the server is capable of providing has been delivered.

In the case of a continuous waveform request, the connection will persist indefinitely. A continuous connection will terminate only from I/O errors, internal implementation errors (i.e. bugs in the client or server code) or external intervention (e.g. server reboot, etc).

While it is not uncommon for a continuous connection to persist for months or longer, it will eventually fail from one of these causes. When it does fail, the server will close its connection and destroy any state associated with the connection. It is up to the client to decide to reconnect, and if so, what streams and start times should be requested. The ISI Toolkit referred to at the start of this document includes an API that takes care of the bookkeeping and other drudgery associated with reestablishing a failed connection. This allows the high level application to loop around calls to a function which reads the next packet from the server and not concern itself with handling transient I/O errors.

ISI_GENERIC_TS (type 1012)

The waveform data are delivered as ISI_GENERIC_TS frames, which are IACP frames with payload identifier 1012. This is a variable length frame. It has a fixed length 64-byte header followed by a variable amount of data, as show below.

Offset	Data type	Length	Description
0	ISI_STREAM_NAME	12	stream name
12	ISI_SRATE	4	nominal sample rate
16	ISI_TSTAMP	10	time of first datum
26	ISI_TSTAMP	10	time of last datum
36	ISI_CD_STATUS	16	CD 1.1 channel status
52	UNIT32	4	number of samples
56	UINT32	4	number of bytes, *N*
60	ISI_DATUM_DESC	4	datum descriptor
64	UINT8	*N*	data

If the *type* field of the datum descriptor is greater than 10, then the data field is the entire raw digitizer packet, otherwise it is the waveform data.